Appendix 1

```
Sub fitfinder()
'
' fitfinder Macro
' Macro recorded 9/29/00 by Michael J. Corey
'
' Keyboard Shortcut: Ctrl+Shift+A
'
Rem initialize rate and correlation arrays
Rem
Rem Fit (rate) array, allowing for future expansion
Dim fit_ar(5000, 96)
Rem Correlation array, which is searched for the best correlation later
Dim cor_ar(1000, 96)
temp1 = 1
Rem get control data from EGG file
tp_no = Range("A4")
sample_no = Range("A3")
Rem This initializes the time array
For ix = 61 To 60 + tp_no
Cells(ix, 1).Value = Cells(ix - 48, 1)
Next ix
Rem
Rem This begins the outer loop. Each "type" of fit (e.g.,
Rem the first three points, or points 7-12) is done for all
Rem samples and the results are stuffed into the rate and
Rem correlation arrays. Then we go back and do the next "type"
Rem of fit.
Rem
temp2 = 0
For stix = 13 To (tp_no + 12)
Rem Initialize the number of points to fit to 3, since there
```

```
Rem is little point in getting a linear fit of 2 points.
pts_to_fit = 4
Rem
Rem This inner loop controls the endpoint of the fit.
Rem
For eix = (stix + 3) To (tp_no + 12)
Rem
Rem This loop repeats the fit for each data-set (i.e., sample).
Rem
temp2 = temp2 + 1
For sample_ix = 2 To sample_no + 1
Rem This copies the data from the raw data matrix to a fixed position.
Rem
Range("B60").Select
For pts_ix = 1 To pts_to_fit
ActiveCell.Offset(1, 0).Select
ActiveCell.Value = Cells(stix + pts_ix - 1, sample_ix)
Next pts_ix
Rem The row index into the arrays is calculated beforehand
Rem to make coding easier.
Rem
Rem Case routine on number of points to fit
Rem
If pts_to_fit = 4 Then GoTo fit4
If pts_to_fit = 5 Then GoTo fit5
If pts_to_fit = 6 Then GoTo fit6
If pts_to_fit = 7 Then GoTo fit7
If pts_to_fit = 8 Then GoTo fit8
If pts_to_fit = 9 Then GoTo fit9
If pts_to_fit = 10 Then GoTo fit10
fit4:
```

```
Range("B2").FormulaR1C1 = "=LinEst(r61c2:r64c2,r61c1:r64c1)"
Range("C2").FormulaR1C1 = "=correl(r61c2:r64c2,r61c1:r64c1)"
GoTo fitend
fit5:
Range("B2").FormulaR1C1 = "=LinEst(r61c2:r65c2,r61c1:r65c1)"
Range("C2").FormulaR1C1 = "=correl(r61c2:r65c2,r61c1:r65c1)"
GoTo fitend
fit6:
Range("B2").FormulaR1C1 = "=LinEst(r61c2:r66c2,r61c1:r66c1)"
Range("C2").FormulaR1C1 = "=correl(r61c2:r66c2,r61c1:r66c1)"
GoTo fitend
fit7:
Range("B2").FormulaR1C1 = "=LinEst(r61c2:r67c2,r61c1:r67c1)"
Range("C2").FormulaR1C1 = "=correl(r61c2:r67c2,r61c1:r67c1)"
GoTo fitend
fit8:
Range("B2").FormulaR1C1 = "=LinEst(r61c2:r68c2,r61c1:r68c1)"
Range("C2").FormulaR1C1 = "=correl(r61c2:r68c2,r61c1:r68c1)"
GoTo fitend
fit9:
Range("B2").FormulaR1C1 = "=LinEst(r61c2:r69c2,r61c1:r69c1)"
Range("C2").FormulaR1C1 = "=correl(r61c2:r69c2,r61c1:r69c1)"
GoTo fitend
fit10:
Range("B2").FormulaR1C1 = "=LinEst(r61c2:r70c2,r61c1:r70c1)"
Range("C2").FormulaR1C1 = "=correl(r61c2:r70c2,r61c1:r70c1)"
GoTo fitend
fitend:
fit_ar(temp1, sample_ix - 1) = Range("B2")
cor_ar(temp1, sample_ix - 1) = Range("C2")
Rem The next four lines display interim data in various parts of
```

```
Rem the spreadsheet, mostly to make debugging easier.
temp3 = Range("B2")
temp4 = Range("C2")
Cells(temp2 + 30, sample_ix) = Range("B2").Value
Cells(temp2 + 90, sample_ix) = Range("C2").Value
Next sample_ix
Rem increment the fit counter
temp1 = temp1 + 1
Rem increment the length of range to fit
pts_to_fit = pts_to_fit + 1
Rem current limit is ten points in the fit range
If pts_to_fit = 11 Then GoTo endthiseix
dumberlabel:
Next eix
Next stix
GoTo skipend
endthiseix:
   eix = tp_no + 13
   GoTo dumberlabel
Rem
Rem Now the routine that determines the best fit.
Rem I will scan from the end of the array so that if there
Rem is a tie, it will pick up the run with the most data points.
Rem
skipend:
Cells(8, 1).Value = "max correl"
Cells(9, 1).Value = "rate"
Cells(10, 1).Value = "notes"
For sample_ix = 1 To sample_no
no_of_fits = (tp_no - 2) * (tp_no - 3) / 2
Rem Note: the previous calculation assumes the fit begins with 4 points.
```

```
Rem In the current code the maximum size of fit is 10 points, and this
Rem formula doesn't take that into account--it would have to be changed
Rem to accommodate fits of more than 10 points, but the principle would
Rem be the same.
equalflag = 0
Dim correlmax As Variant
Dim tempcorr As Variant
correlmax = -1.1
fitmax = -1
For fitix = no_of_fits To 1 Step -1
tempcorr = cor_ar(fitix, sample_ix)
Rem Note: the correl function can yield a DIV/0 error
Rem when the correlation is somehow strange, for example
Rem in negative-control reactions.
Rem The next statement skips the cases in which the error
Rem appears.
If IsNumeric(tempcorr) Then
    GoTo nexter1
    Else: GoTo nexter
End If
nexter1:
If tempcorr = correlmax Then equalflag = 1
If tempcorr > correlmax Then GoTo prenexter
GoTo nexter
prenexter:
equalflag = 0
fitmax = fitix
correlmax = tempcorr
Rem This next complicated routine deconvolutes the triangle number representing
Rem the mapping between the index into the fit array and the actual time range that
Rem was used--for example, using 9 timepoints, a fitix of 3 would be the third fit,
```

```
Rem which would be timepoints 1 to 6; a fitix of 10 would be timepoints 2 to 8.
Rem obviously this routine can be made smaller by having some kind of "for" loop
Rem instead of all the if statements. That would also allow it to handle more than
Rem ten timepoints, which is currently the limit of the whole program.
If fitix > tp_no - 3 Then GoTo tryx2
fitlo = 1
fithi = fitix + 3
GoTo nexter
tryx2:
If fitix > 2 * tp_no - 7 Then GoTo tryx3
fitlo = 2
fithi = fitix + 7 - tp_no
GoTo nexter
tryx3:
If fitix > 3 * tp_no - 12 Then GoTo tryx4
fitlo = 3
fithi = fitix + 12 - tp_no * 2
GoTo nexter
tryx4:
If fitix > 4 * tp_no - 18 Then GoTo tryx5
fitlo = 4
fithi = fitix + 18 - tp_no * 3
GoTo nexter
tryx5:
If fitix > 5 * tp_no - 25 Then GoTo tryx6
fitlo = 5
fithi = fitix + 25 - tp_no * 4
GoTo nexter
tryx6:
If fitix > 6 * tp_no - 33 Then GoTo tryx7
fitlo = 6
```

```
fithi = fitix + 33 - tp_no * 5
GoTo nexter
tryx7:
If fitix > 7 * tp_no - 42 Then GoTo tryx8
fitlo = 7
fithi = fitix + 42 - tp_no * 6
GoTo nexter
tryx8:
If fitix > 8 * tp_no - 52 Then GoTo tryx9
fitlo = 8
fithi = fitix + 52 - tp_no * 7
GoTo nexter
tryx9:
If fitix > 9 * tp_no - 63 Then Stop
fitlo = tr_no - 11
fithi = tp_no - (fitix - 37)
GoTo nexter
nexter:
Next fitix
Rem Display the hard-won best correlation, associated rate, and time range used.
Cells(8, sample_ix + 1) = correlmax
Cells(9, sample_ix + 1) = fit_ar(fitmax, sample_ix)
Cells(10, sample_ix + 1) = "fit " + CStr(fitlo) + " to " + CStr(fithi)
Next sample_ix
End Sub
```

Appendix 2

241.3
1
9
9
MJC5.RP[illegible] 20.11.2000 14:15:06
2.01 1.24 3

Time	A3	A4	A5	B3	B4	B5	C3	C4	C5
0	142337	198427	282680	10882	17286	50090	120279	137692	136200
31.2	154969	222070	338398	11467	18024	51071	119729	136967	135436
61.3	166526	244896	388966	11583	18137	51124	118809	135772	134596
91.3	177288	266025	435282	11657	18217	51120	117688	134883	133384
121.3	187991	286313	478786	11824	18186	50952	116941	133818	132023
151.3	198051	306084	519934	11978	18296	50721	115978	132570	131284
181.3	208351	325526	557892	12006	18340	50636	115283	131989	130199
211.3	218437	343973	593623	12062	18326	50507	114468	130781	129117
241.3	228325	362271	627305	12259	18279	50344	114006	130131	128443

Appendix 3

241.3		
1	-31.614891	-0.995236
9		
9		
MJC5.RPT	20.11.2000	14:15:06
2.01	1.24	3

max correl	0.9999951	0.999997	0.99984177	0.9902173	0.956535	0.8042586	-0.98781	-0.98966	-0.99095
rate	331.93168	652.21619	1435.7657	4.1695772	1.919422	10.411267	-28.554	-28.023	-34.3462
notes	fit 6 to 9	fit 4 to 7	fit 3 to 6	fit 2 to 6	fit 2 to 7	fit 1 to 4	fit 1 to 4	fit 6 to 9	fit 1 to 5

Time	A3	A4	A5	B3	B4	B5	C3	C4	C5
0	142337	198427	282680	10882	17286	50090	120279	137692	136200
31.2	154969	222070	338398	11467	18024	51071	119729	136967	135436
61.3	166526	244896	388966	11583	18137	51124	118809	135772	134596
91.3	177288	266025	435282	11657	18217	51120	117688	134883	133384
121.3	187991	286313	478786	11824	18186	50952	116941	133818	132023
151.3	198051	306084	519934	11978	18296	50721	115978	132570	131284
181.3	208351	325526	557892	12006	18340	50636	115283	131989	130199
211.3	218437	343973	593623	12062	18326	50507	114468	130781	129117
241.3	228325	362271	627305	12259	18279	50344	114006	130131	128443

383.01808	742.25889	1672.7546	8.066686	9.607484	10.411267	-28.554	-31.6295	-30.524
375.45678	726.00556	1616.26211	6.8755017	6.621386	5.9133514	-28.7757	-32.4664	-34.3462
367.59442	711.83165	1565.31615	6.2845871	5.339885	2.6869172	-29.3101	-34.0004	-34.0775
362.18593	699.95175	1515.08105	5.5029586	4.466975	0.9391404	-28.8491	-32.9962	-34.203
357.769	687.9326	1467.2062	4.8744994	3.69283	-0.226583	-28.5167	-33.178	-34.41
353.8321	677.13031	1421.93685	4.6925229	2.98143	-1.117821	-27.4908	-32.4995	-33.7145
361.30767	703.59187	1538.10089	3.764263	1.870181	-1.174205	-31.2029	-34.0106	-37.6282
355.60397	692.02935	1496.52598	4.1695772	1.962817	-2.866657	-30.9569	-35.5059	-35.9209
352.21878	682.46681	1451.78988	3.8289985	1.919422	-3.351375	-29.7838	-33.6553	-35.4619
349.2975	671.84166	1407.96718	3.4978006	1.650081	-3.626635	-29.0835	-33.6925	-35.3611
346.44278	662.26498	1365.9922	3.6133597	1.27085	-3.891538	-27.6568	-32.7119	-34.2507
346.31774	670.59796	1435.7657	4.4406335	1.467754	-4.513374	-30.4087	-35.0756	-37.1803
344.94916	665.11777	1396.06535	3.8544857	1.601985	-4.53421	-28.9595	-32.6337	-36.0042
343.31575	656.42933	1355.58138	3.4037896	1.348177	-4.507104	-28.2868	-33.0073	-35.6729
341.26961	648.35926	1316.25617	3.5779737	0.946909	-4.595845	-26.6744	-31.9358	-34.1494
339.64045	652.21619	1345.55402	3.9581084	1.56841	-5.53882	-26.8971	-32.6838	-33.8872
339.14603	644.59517	1307.80713	3.2838643	1.226795	-5.097408	-26.7511	-33.1527	-34.2299
337.6527	637.48812	1270.7113	3.5532585	0.73285	-5.017277	-25.1032	-31.6635	-32.6651
334.29926	632.98969	1258.37217	2.4486011	1.534317	-4.679316	-26.7022	-31.8953	-32.2178
333.82595	627.07719	1225.00586	3.1524772	0.721154	-4.728085	-24.3964	-30.2849	-30.8044
331.93168	615.18674	1177.2935	2.9472259	-0.20762	-4.139197	-22.1495	-28.023	-31.6149

0	131284
31.2	130199
61.3	129117
91.3	128443
121.3	128443
151.3	128443
181.3	128443
211.3	128443
241.3	128443

0.999633	0.9998448	0.99947062	0.8986993	0.882256	0.8042586	-0.98781	-0.9954	-0.99285
0.9996012	0.9996634	0.99909356	0.9144503	0.813995	0.6398954	-0.99386	-0.99746	-0.99095
0.9994528	0.9995317	0.99873209	0.9339036	0.809147	0.37913	-0.99637	-0.9973	-0.99465
0.9994593	0.9994544	0.99823055	0.9309838	0.806686	0.1635285	-0.99743	-0.99743	-0.99665
0.9994765	0.9993158	0.99767264	0.9267276	0.787346	-0.046029	-0.99811	-0.99827	-0.99775
0.9994797	0.999207	0.99708114	0.9416045	0.746819	-0.24955	-0.99695	-0.99822	-0.99786
0.9999576	0.999843	0.99968876	0.9877688	0.866925	-0.574589	-0.99746	-0.99888	-0.99368
0.9998471	0.9997762	0.99944197	0.9902173	0.931044	-0.80943	-0.99866	-0.99858	-0.99535
0.9998482	0.9997355	0.99902011	0.9881171	0.956535	-0.894652	-0.99813	-0.99706	-0.99714
0.9998444	0.9996188	0.99853484	0.9837782	0.94338	-0.935272	-0.99828	-0.99816	-0.99819
0.999826	0.9995276	0.99801335	0.988755	0.870883	-0.956765	-0.99607	-0.99779	-0.99766
0.9999425	0.9999801	0.99984177	0.9869261	0.863685	-0.932756	-0.99774	-0.99648	-0.99391
0.9999627	0.9999553	0.99950457	0.979931	0.93197	-0.964636	-0.99748	-0.99512	-0.99617
0.9999631	0.9998555	0.99909758	0.9739874	0.924294	-0.97896	-0.9981	-0.99717	-0.9977
0.9999464	0.999777	0.99865159	0.98309	0.811485	-0.986851	-0.99561	-0.99716	-0.99676
0.9999861	0.999997	0.99975498	0.9659756	0.869944	-0.987001	-0.99804	-0.99116	-0.99463
0.9999919	0.9999274	0.99944582	0.9554568	0.862655	-0.988493	-0.99898	-0.99554	-0.99728
0.9999821	0.9998735	0.99908896	0.973789	0.679448	-0.992962	-0.99645	-0.99571	-0.99675
0.9999258	0.9999557	0.99973904	0.9436501	0.862855	-0.980615	-0.9985	-0.99164	-0.99548
0.9999614	0.999932	0.99948407	0.9586433	0.570086	-0.99023	-0.99459	-0.99388	-0.99644
0.9999951	0.9999875	0.99983614	0.9126798	-0.29348	-0.989244	-0.99499	-0.98966	-0.99524